AMENDMENTS TO THE CLAIMS

1.   (Currently Amended)   A method of simulating an instruction set architecture

(ISA) with a instruction set simulator (ISS), comprising:

fetching a decoded instruction during a run time, wherein the decoded instruction is

decoded from an original instruction in a target application program during a compile

time preceding the run time, the decoded instruction pointing to a template

configured to implement the functionality of the original instruction, wherein the

template is associated with an instruction class that describes a set of instructions of

the instruction set architecture having a common behavior, and wherein the original

instruction is contained in the instruction class;

determining whether the decoded instruction fetched during a run time is modified from the

original instruction; and

executing the template if the decoded instruction was not modified from the original

instruction.

2.   (Previously Presented)   The method of claim 1, further comprising decoding the

original instruction by selecting the template corresponding to the original instruction

and customizing the template based on the data in original instruction prior to

fetching the decoded instruction.

3.   (Currently Amended)   The method of claim 2, ~~wherein the template~~

~~corresponds to a first class of one or more instructions and~~ wherein the template has

a corresponding mask usable to identify instructions belonging to the ~~first~~ <u>instruction</u> class.

4. (Original) The method of claim 3, wherein selecting a template comprises: comparing the original instruction to the mask corresponding to the template; and selecting the template if the mask matches the original instruction.

5. (Original) The method of claim 2, wherein customizing the template comprises determining a value of a parameter in the template based on the data in the original instruction.

6. (Previously Presented) The method of claim 3, wherein customizing the template comprises determining a value of a parameter in the template based on the data in the original instruction.

7. (Original) The method of claim 6, further comprising compiling a first program comprising the customized template in the compile time.

8. (Original) The method of claim 7, further comprising optimizing the template during the compile time.

9. (Original) The method of claim 1, further comprising:

re-decoding the fetched instruction during the run time if the fetched instruction was

modified, wherein the re-decoded instruction designates a function configured to

implement the functionality of the instruction; and

executing the designated function if the instruction was modified.

10.   (Original)   The method of claim 1, further comprising executing the modified

instruction using an interpretive process.

11.   (Original)   The method of claim 8, further comprising compiling the target

application program to generate the original instruction.

12.   (Currently Amended)   A <u>system,</u> ~~generic instruction model stored on a computer~~

~~readable medium, which when read by a processor is~~ for use in a instruction set

~~architecture (ISA) simulator, the generic instruction model~~ comprising:

<u>a processor; and</u>

<u>a computer readable medium having stored thereon a generic instruction model, the</u>

<u>generic instruction model executable by the processor in an instruction set simulator,</u>

<u>the generic instruction model comprising</u> an instruction specification usable to

interpret each instruction in an ISA, the instruction specification comprising one or

more operation classes;

wherein each operation class ~~defining~~ defines a set of one or more instructions, the

operation class having an operation mask usable to identify instructions belonging to

the class; and

~~further~~ wherein the operation class comprises one or more symbols and an expression

describing the class in terms of the one or more symbols, each symbol having a

corresponding set of one or more symbol types, each symbol type in the set

comprising information usable to determine the symbol when compared to an

instruction; and

wherein the instruction set simulator is configured to perform

fetching a decoded instruction during a run time, wherein the decoded instruction is

decoded from an original instruction in a target application program during a

compile time preceding the run time, the decoded instruction pointing to a

template configured to implement the functionality of the original instruction,

wherein the template is associated with an operation class, and wherein the

original instruction is contained in the operation class;

determining whether the decoded instruction fetched during a run time is modified

from the original instruction; and

executing the template if the decoded instruction was not modified from the original

instruction.

13. (Currently Amended) The system ~~model~~ of claim 12, wherein the set of instructions has a common behavior and the expression defines the behavior of the class in terms of the one or more symbols.

14. (Currently Amended) The system ~~model~~ of claim 12, wherein one symbol type in the type set is an constant type.

15. (Currently Amended) The system ~~model~~ of claim 14, wherein the type set comprises a plurality of constant types, each constant type having a corresponding type mask usable to determine the constant when compared to an instruction.

16. (Currently Amended) The system ~~model~~ of claim 12, wherein one symbol type in the type set is a register type.

17. (Currently Amended) The system ~~model~~ of claim 16, wherein the register type comprises a register index and a register class.

18. (Currently Amended) The system ~~model~~ of claim 12, wherein one symbol type in the type set is an operation type.

19. (Currently Amended) The system model of claim 18, wherein the type set comprises a plurality of operation types, each operation type having a corresponding type mask usable to determine the operation when compared to an instruction.

20. (Currently Amended) The system model of claim 12, wherein at least one operation class comprises a plurality of expressions, each expression being conditional on data within an instruction.

21. (Currently Amended) The system model of claim 12, wherein each instruction comprises a series of slots, each slot comprising data translatable into an operation.

22. (Currently Amended) The system model of claim 12, wherein each instruction comprises a series of binary data values and the operation mask comprises a series of mask positions wherein each mask position corresponds to one instance of a binary data value.

23. (Currently Amended) The system model of claim 12, wherein each mask position has a value selected from a group comprising: a binary one value, a binary zero value and a do not care value.

24.   (Currently Amended)   A computer readable medium having stored thereon a

set of instructions executable by a machine to perform operations for simulating an

instruction set architecture (ISA), said operations comprising:

fetching a decoded instruction during a run time, wherein the decoded instruction is

decoded from an original instruction in a target application program during a compile

time preceding the run time, the decoded instruction pointing to a template

configured to implement the functionality of the original instruction, wherein the

template is associated with an instruction class that describes a set of instructions of

the instruction set architecture having a common behavior, and wherein the original

instruction is contained in the instruction class;

determining whether the decoded instruction fetched during a run time is modified from the

original instruction; and

executing the template if the decoded instruction was not modified from the original

instruction.

25.   (Original)   The computer readable medium of claim 24, wherein the template

corresponds to a first class of one or more instructions and wherein the template has

a corresponding mask usable to identify instructions belonging to the first class.

26.   (Original)   The computer readable medium of claim 24, further comprising:

re-decoding the fetched instruction during the run time if the fetched instruction was

modified, wherein the re-decoded instruction designates a function configured to

implement the functionality of the instruction; and

executing the designated function if the instruction was modified.

27.   (Original)   The computer readable medium of claim 24, further comprising

executing the modified instruction using an interpretive process.